

# Flash memory

From Wikipedia, the free encyclopedia

**Flash memory** is non-volatile computer memory that can be electrically erased and reprogrammed. It is a technology that is primarily used in memory cards. Unlike EEPROM, it is erased and programmed in blocks consisting of multiple locations (in early flash the entire chip had to be erased at once). Flash memory costs far less than EEPROM and therefore has become the dominant technology wherever a significant amount of non-volatile, solid-state storage is needed. Examples of applications include digital audio players, digital cameras and mobile phones. Flash memory is also used in USB flash drives (thumb drives, handy drive), which are used for general storage and transfer of data between computers. It has also gained some popularity in the game console market, where it is often used instead of EEPROMs or battery-powered static RAM (SRAM) for game save data.

## Contents

- 1 Overview
- 2 Principles of operation
- 3 History
- 4 Limitations
- 5 Low-level access
  - 5.1 NOR memories
  - 5.2 NAND memories
- 6 Serial flash
- 7 Flash file systems
- 8 Capacity
- 9 Speed
- 10 Data corruption and recovery
- 11 Flash memory as a replacement for hard dr.
- 12 See also
- 13 External links
- 14 References

## Overview

Flash memory is non-volatile, which means that it does not need power to maintain the information stored in the chip. In addition, flash memory offers fast read access times (although not as fast as volatile DRAM memory used for main memory in PCs) and better kinetic shock resistance than hard disks. These characteristics explain the popularity of flash memory for applications such as storage on battery-powered devices. Another allure of flash memory is that when packaged in a "memory card", it is enormously durable, being able to withstand intense pressure, extremes of temperature and immersion in water<sup>1</sup>.

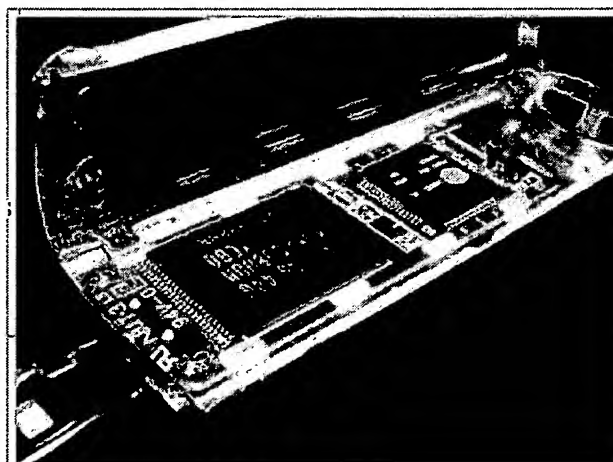
### Memory types

#### Volatile

- DRAM
- eDRAM
- SRAM
- 1T-SRAM
- Upcoming
  - Z-RAM
  - TTRAM

#### Non-Volatile

- Flash memory
- ROM
  - PROM
  - EPROM
  - EEPROM
- Upcoming
  - FeRAM
  - MRAM
  - PRAM
  - SONOS
  - RRAM
  - NRAM



A USB flash drive. The chip on the left is the flash memory. The microcontroller is on the right.

## Principles of operation

Flash memory stores information in an array of floating gate transistors, called "cells", each of which traditionally stores one bit of information. Newer flash memory devices, sometimes referred to as multi-level cell devices, can store more than 1 bit per cell, by using more than two levels of electrical charge, placed on the floating gate of a cell.

In NOR gate flash, each cell looks similar to a standard MOSFET, except that it has two gates instead of just one. One gate is the control gate (CG) like in other MOS transistors, but the second is a floating gate (FG) that is insulated all around by an oxide layer. The FG is between the CG and the substrate. Because the FG is isolated by its insulating oxide layer, any electrons placed on it get trapped there and thus store the information. When electrons are on the FG, they modify (partially cancel out) the electric field coming from the CG, which modifies the threshold voltage ( $V_t$ ) of the cell. Thus, when the cell is "read" by placing a specific voltage on the CG, electrical current will either flow or not flow, depending on the  $V_t$  of the cell, which is controlled by the number of electrons on the FG. This presence or absence of current is sensed and translated into 1s and 0s, reproducing the stored data. In a multi-level cell device, which stores more than 1 bit of information per cell, the *amount* of current flow will be sensed, rather than simply detecting presence or absence of current, in order to determine the number of electrons stored on the FG.

A NOR flash cell is programmed (set to a specified data value) by starting up electrons flowing from the source to the drain, then a large voltage placed on the CG provides a strong enough electric field to suck them up onto the FG, a process called hot-electron injection. To erase (reset to all 1s, in preparation for reprogramming) a NOR flash cell, a large voltage differential is placed between the CG and source, which pulls the electrons off through quantum tunneling. In single-voltage devices (virtually all chips available today), this high voltage is generated by an on-chip charge pump. Most modern NOR flash memory components are divided into erase segments, usually called either blocks or sectors. All of the memory cells in a block must be erased at the same time. NOR programming, however, can generally be performed one byte or word at a time.

NAND gate flash uses tunnel injection for writing and tunnel release for erasing. NAND flash memory forms the core of the removable USB interface storage devices known as USB flash drives.

As manufacturers increase the density of flash devices, individual cells shrink and the number of electrons in any cell becomes very small. Coupling between adjacent floating gates can change the cell write characteristics. New designs, such as charge trap flash, attempt to provide better isolation between adjacent cells.

## History

Flash memory (both NOR and NAND types) was invented by Dr. Fujio Masuoka while working for Toshiba in 1984. According to Toshiba, the name "flash" was suggested by Dr. Masuoka's colleague, Mr. Shoji Ariizumi, because the erasure process of the memory contents reminded him of a flash of a camera. Dr. Masuoka presented the invention at the IEEE 1984 International Electron Devices Meeting (IEDM) held in San Francisco, California. Intel saw the massive potential of the invention and introduced the first commercial NOR type flash chip in 1988.

NOR-based flash has long erase and write times, but has a full address/data (memory) interface that allows random access to any location. This makes it suitable for storage of program code that needs to be infrequently updated, such as a computer's BIOS or the firmware of set-top boxes. Its endurance is 10,000 to 1,000,000 erase cycles. NOR-based flash was the basis of early flash-based removable media; CompactFlash was originally based on it, though later cards moved to the less expensive NAND flash.

NAND flash, which Toshiba announced at ISSCC in 1989, followed. It has faster erase and write times, higher density, and lower cost per bit than NOR flash, and ten times the endurance. However its I/O interface allows only sequential access to data. This makes it suitable for mass-storage devices such as PC cards and various memory cards, and somewhat less useful for computer memory. The first NAND-based removable media format was SmartMedia, and numerous others have followed: MultiMediaCard, Secure Digital, Memory Stick and xD-Picture Cards. A new generation of these formats is becoming a reality with RS-MMC (Reduced Size MultiMediaCard), the micro- and miniSD variants of Secure Digital and the new USB/Memory card hybrid Intelligent Stick. The new formats exhibit a greatly reduced size, usually under 4 cm<sup>2</sup>.

## Limitations

One limitation of flash memory is that although it can be read or programmed a byte or a word at a time in a random access fashion, it must be erased a "block" at a time. This generally sets all bits in the block to 1. Starting with a freshly erased block, any location within that block can be programmed. However, once a bit has been set to 0, only by erasing the entire block can it be changed back to 1. In other words, flash memory (specifically NOR flash) offers random-access read and programming operations, but cannot offer arbitrary random-access rewrite or erase operations. A location can, however, be rewritten as long as the new value's 0 bits are a superset of the over-written value's. For example, a nibble value may be erased to 1111, then written as 1110. Successive writes to that nibble can change it to 1010, then 0010, and finally 0000. Although data structures in flash memory can not be updated in completely general ways, this allows members to be "removed" by marking them as invalid. This technique must be modified somewhat for multi-level devices, where one memory cell holds more than one bit.

Another limitation is that flash memory has a finite number of erase-write cycles (most commercially available flash products are guaranteed to withstand 1 million programming cycles). This effect is partially offset by some chip firmware or file system drivers by counting the writes and dynamically remapping the blocks in order to spread the write operations between the sectors. This technique is called wear levelling. Another mechanism is to perform write verification and remapping to spare sectors in case of write failure, which is named bad block management (BBM).

## Low-level access

Low-level access to a physical flash memory by device driver software is different from accessing common memories. Whereas a common RAM will simply respond to read and write operations by returning the contents or altering them immediately, flash memories need special considerations, especially when used as program memory akin to a read-only memory (ROM).

While reading data can be performed on individual addresses on NOR memories unlocking (making available for erase or write), erasing and writing operations are performed block-wise on all flash

memories. A typical block size will be 64, 128, or 256 KiB. This cannot be done with NAND memories.

One group called Open NAND flash Interface Working Group (<http://www.onfi.org/>) (ONFI) aims to develop a standardized low-level NAND flash interface that allows interoperability between NAND devices from various vendors. The goals of this group include developing a standardized chip-level interface (pin-out) for NAND flash, a standard command set and a self identification mechanism (à la SDRAM's SPD EEPROM). The specification was released on January 22, 2007.

## NOR memories

The read-only mode of NOR memories is similar to reading from a common memory, provided address and data bus is mapped correctly, so NOR flash memory is much like any address-mapped memory. NOR flash memories can be used as execute in place (XIP) memory, meaning it behaves as a ROM memory mapped to a certain address. NOR flash memories have no intrinsic bad block management, so when a flash block is worn out, either the software using it has to handle this, or the device breaks.

When unlocking, erasing or writing NOR memories, special commands are written to the first page of the mapped memory. These commands are defined as the *Common Flash memory Interface (CFI)* (defined by Intel) and the flash circuit will provide a list of all available commands to the physical driver.

Apart from being used as a ROM, the NOR memories can also be partitioned with a file system and used as any storage device. However, NOR file systems are typically very slow to write when compared with NAND file systems.

## NAND memories

NAND flash architecture was introduced by Toshiba in 1989. NAND flash memories cannot provide execute in place due to their different construction principles. These memories are accessed much like block devices such as hard disks or memory cards. The pages are typically 512 or 2,048 bytes in size. Associated with each page are a few bytes (typically 12–16 bytes) that should be used for storage of an error detection and correction checksum.

The pages are typically arranged in blocks. A typical block would be 32 pages of 512 bytes or 64 pages of 2048 bytes.

While programming is performed on a page basis, erasure can only be performed on a block basis.

NAND devices typically have software-based bad block management. This means that when a logical block is accessed it is mapped to a physical block, and the device has a number of blocks set aside for compensating bad blocks and for storing primary and secondary mapping tables.

The error-correcting and detecting checksum will typically correct an error where one bit per 256 bytes is incorrect. When this happens, the block is marked bad in a logical block allocation table, and its undamaged contents are copied to a new block and the logical block allocation table is altered accordingly. If more than one bit in the memory is corrupted, the contents are partly lost, i.e. it is no longer possible to reconstruct the original contents.

Most NAND devices are shipped from the factory with some bad blocks which are typically identified

and marked according to a specified bad block marking strategy. By allowing some bad blocks, the manufacturers achieve far higher yields than would be possible if all blocks were tested good. This significantly reduces NAND flash costs and increases the size of the parts.

The first error-free physical block (block 0) is always guaranteed to be readable and free from errors. Hence, all vital pointers for partitioning and bad block management for the device must be located inside this block (typically a pointer to the bad block tables etc). If the device is used for booting a system, this block may contain the master boot record.

When executing software from NAND memories, virtual memory strategies are used: memory contents must first be paged or copied into memory-mapped RAM and executed there. A memory management unit (MMU) in the system is helpful, but this can also be accomplished with overlays. For this reason, some systems will use a combination of NOR and NAND memories, where a smaller NOR memory is used as software ROM and a larger NAND memory is partitioned with a file system for use as a random access storage area. NAND is best suited to flash devices requiring high capacity data storage. This type of flash architecture offers storage space up to 512-MB and has faster erase, write, and read capabilities over NOR architecture.

## Serial flash

Serial flash is a small, low-power flash memory that uses a serial interface, typically SPI, for sequential data access. Serial flash requires fewer wires on the printed circuit board (PCB) than parallel flash memories to transfer data. A reduction in board space, power consumption and system cost are some of the benefits of the lower pin-count interface.

A saving of pins translates into multiple cost reductions. Many ASIC/controller designs are pad-limited. In many designs the number of bond pads, rather than the amount of gates used for the core and logic, dictates the size of the die. Eliminating bond pads allows for a more compact ASIC/controller design that results in a reduced die size, which lowers the die cost and increases the die per wafer count. Additionally, reducing the number of active pins allows lower pin-count packages and reductions in assembly and package costs. Of course, the package size of the flash device itself also drastically changes when going from large parallel flash to serial flash. With smaller and lower pin-count packages come reduced PCB area and simplified routing, both of which help lower system costs.

As CPU performance increases, the access times (45 ns+) of traditional parallel flash are not fast enough to directly execute the program code. At the same time embedded SRAM technology allows sub-10 ns access times and DDR2 allows 20 ns access times. Slowness of the flash makes "code shadowing" - storing code in the RAM - inevitable in many devices. In many instances it is still more cost-effective to double the SDRAM density rather than doubling the flash density because of SDRAM's lower cost-per-bit factor and keep the code compressed on the flash.

Among typical applications are firmware storage for hard drives, Ethernet controllers, DSL modems, wireless modems, and so on. In these systems the code is shadowed in the RAM. After the system powers up, the ASIC simply selects the serial flash, sends it one command to start reading the memory, and then continues to clock the serial flash until all of the necessary code has been output. The serial flash implements "bulk read" mode and incorporates an internal address counter so that on every clock cycle the flash device outputs the next bit of data.

The industry's average speeds for serial buses are 50 MHz. These devices are capable of sustaining read throughputs at 50 Mbps, or 5 MB per second. With such throughputs, an entire 64-Mbit device can be read in less than two seconds.

## Flash file systems

Because of the particular characteristics of flash memory, it is best used with specifically designed file systems which spread writes over the media and deal with the long erase times of NOR flash blocks. The basic concept behind flash file systems is: When the flash store is to be updated, the file system will write a new copy of the changed data over to a fresh block, remap the file pointers, then erase the old block later when it has time. One of the earliest flash file systems was Microsoft's FFS2 (presumably preceded by FFS1), for use with MS-DOS in the early 1990s. Around 1994, the PCMCIA industry group approved the FTL (Flash Translation Layer) specification, which allowed a flash device to look like a FAT disk, but still have effective wear levelling. Other commercial systems such as FlashFX by Datalight were created to avoid patent concerns with FTL.

JFFS was the first flash-specific file system for Linux, but it was quickly superseded by JFFS2, originally developed for NOR flash. Then YAFFS was released in 2003, dealing specifically with NAND flash, and JFFS2 was updated to support NAND flash too. In practice, these filesystems are only used for "Memory Technology Devices" ("MTD"), which are embedded flash memories which do not have a controller. Removable flash media, such as SD and CF cards and USB flash drives, have a controller (often built into the card) to perform wear-levelling and error correction, so use of JFFS2 or YAFFS does not add any benefit. These removable flash memory devices are often used with the old FAT filesystem for compatibility with cameras and other portable devices. Controllerless removable flash memory devices also exist; For example, SmartMedia is even electrically compatible with the Toshiba TC58 series of NAND flash chips.

## Capacity

Common flash memory parts (individual internal components or "chips") range widely in capacity from kilobits to several gigabits each. Multiple chips are often arrayed to achieve higher capacities for use in devices such as the iPod nano or SanDisk Sansa e200. The capacity of flash chips generally follows Moore's law because they are produced with the same processes used to manufacture other integrated circuits. However, there have also been jumps beyond Moore's law due to innovations in technology.

In 2005, Toshiba and SanDisk developed a NAND flash chip capable of storing 1 gigabyte of data using MLC (multi-level cell) technology, capable of storing 2 bits of data per cell. In September 2005, Samsung Electronics announced that it had developed the world's first 2 gigabyte chip.

In March 2006, Samsung announced flash hard drives with a capacity of 4 gigabytes, essentially the same order of magnitude as smaller laptop hard drives, and in September of 2006, Samsung announced 64 gigabits chips produced using a 40 nm manufacturing process[2] ([http://tgdaily.com/2006/09/11/samsung\\_40nm\\_flash/](http://tgdaily.com/2006/09/11/samsung_40nm_flash/)).

For some flash memory products such as memory cards and USB-memories, as of mid 2006, 256 megabyte and smaller devices have been largely discontinued. 1 GB capacity flash memory has become

the normal storage space for people who do not extensively use flash memory, while more and more consumers are adopting 2 GB or 4 GB flash drives.

Hitachi (formerly the hard disk unit of IBM) has a competing hard-drive mechanism, the Microdrive, that can fit inside the shell of a type II CompactFlash card. It has a capacity up to 8 GB. BiTMicro offers a 155 GB 3.5" Solid-State disk named the "Edisk". [3]  
([http://www.tgdaily.com/2005/09/13/bitmicro\\_rolls\\_out\\_155\\_gig\\_solid/](http://www.tgdaily.com/2005/09/13/bitmicro_rolls_out_155_gig_solid/))

## Speed

Flash memory cards are available in different speeds. Some are specified the approximate transfer rate of the card such as 2 MB per second, 12 MB per second, etc. The exact speed of these cards depends on which definition of "megabyte" the marketer has chosen to use.

Many cards are simply rated 100x, 130x, 200x, etc. For these cards the base assumption is that 1x is equal to 150 kibibytes per second. This was the speed at which the first CD drives could transfer information, which was adopted as the reference speed for flash memory cards. Thus, when comparing a 100x card to a card capable of 12 MiB per second the following calculations are useful:

$150 \text{ KiB} \times 100 = 15000 \text{ KiB per second} = 14.65 \text{ MiB per second.}$

Therefore, the 100x card is 14.65 MiB per second, which is faster than the card that is measured at 12 MiB per second.

## Data corruption and recovery

The most common cause of data corruption is removal of the flash memory device while data is being written to it. The situation is aggravated by the usage of unsuitable file systems that are not designed for removable devices, or that are mounted async (where there is data still waiting to write when the device is removed).

Data recovery from flash memory devices can be achieved in some cases. Heuristic (<http://www.cs.washington.edu/homes/oskin/saveimg.html>) and Brute Force (<http://www.linuxjournal.com/article/8366>) methods are examples of recovery that may yield results for general data on a compact flash card.

## Flash memory as a replacement for hard drives

An obvious extension of flash memory would be as a replacement for hard disk drives. Flash memory does not have the mechanical limitations and latencies of hard drives, so the idea of a solid state drive, or SSD, is attractive when considering speed, noise, power consumption, and reliability.

There remain some aspects of flash-based SSD's that make the idea unattractive. For example, the cost per storage ratio of flash memory remains significantly higher than that of platter-based hard drives. Although this ratio is decreasing rapidly for flash memory, it will take some time for flash memory to catch up to the capacities and affordabilities offered by platter-based storage, but as research and development shifts toward the newer technology, this issue might dissolve.

There is also some concern that the finite number of erase/write cycles of flash memory would render flash memory unable to support an operating system. This seems to be a decreasing issue as warranties on flash-based SSD's are trending to equal or exceed those of current hard drives. [4] (<http://www.storagesearch.com/semico-art1.html>) [5] (<http://www.storagesearch.com/bitmicro-art1.html>)

As of May 24, 2006, South Korean consumer-electronics manufacturer Samsung Electronics had released the first flash-memory based PCs, the Q1-SSD and Q30-SSD, both of which have 32GB SSDs. [6] ([http://www.samsung.com/he/presscenter/pressrelease/pressrelease\\_20060524\\_0000257996.asp](http://www.samsung.com/he/presscenter/pressrelease/pressrelease_20060524_0000257996.asp))

At Las Vegas CES 2007 Summit Taiwanese memory company A-DATA showcased SSD hard disk drives based on Flash technology in capacities of 32 GB, 64 GB and 128 GB.<sup>[1]</sup> Sandisk announced an OEM 32GB 1.8" SSD drive at CES 2007 <sup>[2]</sup>

Rather than entirely replacing the hard drive, hybrid techniques such as hybrid drive and ReadyBoost attempt to combine the advantages of both technologies, using flash as a high-speed cache for files on the disk that are often referenced, but rarely modified, such as application and operation system executable files.

## See also

- CompactFlash
- Wear leveling
- DataFlash

## External links

- Open NAND Flash Interface Working Group (<http://www.onfi.org/index.html>)
- A Nonvolatile Memory Overview (<http://aplawrence.com/Makwana/nonvolmem.html>)
- How Flash Memory Works (<http://computer.howstuffworks.com/flash-memory.htm>)
- SanDisk Flash Memory Plant ([http://www.pma-show.com/2006/corporate/sandisk\\_review/001\\_sandisk\\_flash\\_memory\\_facility.html](http://www.pma-show.com/2006/corporate/sandisk_review/001_sandisk_flash_memory_facility.html))
- What is NAND Flash (<http://www.micron.com/products/nand/definingnand>)
- NAND Flash Applications (<http://www.micron.com/products/nand/usingnand>)

## References

- Digital Memories Survive Extremes (<http://news.bbc.co.uk/2/hi/technology/3939333.stm>)
- Flash memory database ([http://www.letsgodigital.org/en/flash\\_memory\\_cards.html](http://www.letsgodigital.org/en/flash_memory_cards.html))

- ↑ Future of Flash revealed (<http://www.theinquirer.net/default.aspx?article=36841>)
- ↑ [1] (<http://www.sandisk.com/Oem/Default.aspx?CatID=1477>)

Retrieved from "[http://en.wikipedia.org/wiki/Flash\\_memory](http://en.wikipedia.org/wiki/Flash_memory)"

Categories: Articles with unsourced statements since February 2007 | All articles with unsourced

statements | Solid-state computer storage media | Computer memory | Non-volatile memory

- This page was last modified 21:08, 15 March 2007.
- All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.) Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a US-registered 501(c)(3) tax-deductible nonprofit charity.